Color Filter Array Demosaicking Using High-Order Interpolation Techniques With a Weighted Median Filter for Sharp Color Edge Preservation

Jim S. Jimmy Li, *Member, IEEE*, and Sharmil Randhawa, *Member, IEEE*

***Abstract*—Demosaicking is an estimation process to determine missing color values when a single-sensor digital camera is used for color image capture. In this paper, we propose a number of new methods based on the application of Taylor series and cubic spline interpolation for color filter array demosaicking. To avoid the blurring of an edge, interpolants are first estimated in four opposite directions so that no interpolation is carried out across an edge. A weighted median filter, whose filter coefficients are determined by a classifier based on an edge orientation map, is then used to produce an output from the four interpolants to preserve edges. Using the proposed methods, the original color can be faithfully reproduced with minimal amount of color artifacts even at edges.**

***Index Terms*—Color filter array, cubic spline, demosaicking, Taylor series, weighted median filter.**

Fig. 1. An 8 × 8 window of the Bayer pattern.

I. Introduction

COLOR filter array demosaicking is a digital process to obtain full color images from images captured by a single image sensor. A single image sensor does not allow the full red, green and blue color planes to be captured. In other words, the image color is captured in a sub-sampled pattern. The most common array is the Bayer [2] color filter array, shown in Fig. 1. The green color is sampled at twice the rate of the red or blue values because the peak sensitivity of the human visual system lies in the green spectrum [2].

The technique to recover missing colors is known as demosaicking. Most demosaicking methods use an edge-directed interpolation, where the gradient around a pixel is analysed to determine a preferred direction. This is to avoid introducing color artifacts that are caused by interpolation across edges. However, any errors in the gradient estimation may manifest themselves as color artifacts in the output image.

A common assumption in demosaicking is that color differences or hue are locally constant [15] based on the fact there is a high interchannel correlation within an object of an image. However, this assumption is only valid within the boundary of an object and it gets violated across edges. Some methods use a weighted average of the neighboring pixels to interpolate the missing color [29], while other methods use gradient values to select a preferred direction for interpolation [14]. In [12], edge indicators in several directions are used to indicate the likelihood of edges in order to prevent interpolation across them. In [23], the estimated values are weighted averages, with weights determined by the gradient. However, all these methods will produce color artifacts when there are errors in the estimation of weights for the average or the gradients. As a result, the performance of these methods is limited by the accuracy of their estimation algorithms.

In this paper, we are proposing a suite of noniterative demosaicking algorithms to determine the missing color values. The proposed algorithms perform differently for various types of images, and there is no outright winner amongst them. However, all the proposed algorithms will produce minimal artifacts and will outperform other existing demosaicking algorithms for most types of images.

Our proposed methods consist of two stages. In the first stage, an interpolation algorithm is used to determine four estimates of the missing color value for four different directions. A number of highly accurate interpolation methods based on Taylor series and cubic spline are proposed. In the second output stage, a classifier based on an edge orientation map is used to assign weights for a weighted median filter which is used to determine the output. Since an edge has two sides, at least half of the four estimates must be on one side of the edge. In other words, at least two of the four estimates are on the same side of the edge, and by using an edge-preserving weighted median filter to determine the output, our methods have been shown to outperform the other demosaicking methods in many aspects.

The remainder of the paper is organized as follows. Our suite of proposed interpolation methods is presented in Sections II–IV for the green plane, while Section V deals with

Manuscript received July 18, 2007; revised March 26, 2009. First published June 23, 2009; current version published August 14, 2009. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Charles A. Bouman.

The authors are with the School of Computer Science, Engineering and Mathematics, Flinders University, Adelaide SA 5042, Australia (e-mail: jimmy.li@flinders.edu.au; sharmil.randhawa@flinders.edu.au).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TIP.2009.2022291

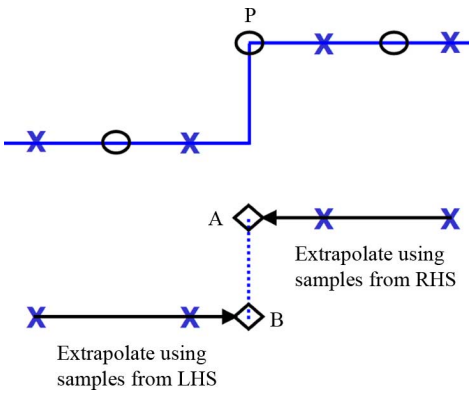

Fig. 2. One-dimensional extrapolation example.

the interpolation for the red and blue planes. Section VI describes the edge orientation classifier and the weighted median filter. Section VII gives the proposed demosaicking algorithm. Section VIII presents the experimental results, and compares the proposed methods with other existing methods, with the conclusion in Section IX.

II. High-Order Extrapolation (HOX)

When there is an edge, only pixels on the same side of the edge should be used for extrapolation. It is crucial to avoid extrapolation across edges, as smooth hue assumptions are not valid across edges. When there is a 1-D edge, as shown in Fig. 2 for example, the circle indicates the position at which the value is to be estimated, and the crosses represent the surrounding known sample values. Point P is the pixel value to be estimated. Interpolation using samples from both sides of the edge will cause blurring. In order to preserve the edge, we apply extrapolation using only samples from either the left-hand side (LHS) or right-hand side (RHS). Point A is the extrapolated value using samples on the RHS of the edge, and point B is another extrapolated value using samples on the LHS of the edge in Fig. 2. In this example, for a 1-D edge, two extrapolated samples, one for each side of the edge, will be determined. However, for a 2-D image, we extrapolate in the four directions towards the center, i.e., from left, right, top and bottom.

In this section, we describe a high-order extrapolation method with high accuracy to determine four directional estimates of the missing color pixel [17]–[21], [28]. Each of the directional estimates is obtained by exploiting the spectral correlations among neighboring pixels along that particular direction. In this way, we will show that the estimates obtained are more accurate than those in [23], due to the inclusion of higher-order terms. The spectral correlations refer to the fact there is a high correlation between the green and red/blue pixel values within a local neighborhood.

Based on the Bayer pattern, each missing green value is surrounded by four known green values located in the left, right, top and bottom directions. Hence, to determine the missing values in the green plane, the estimates for the missing green value are extrapolated in the four same directions. The green plane is extrapolated first as it contains the most samples, twice as many as the red or blue samples. The approximation is carried out using Taylor series [13] as follows:

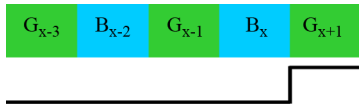

Fig. 3. 1-D Bayer pattern and edge boundary on the RHS of B_x.

$$g(x) = \sum_{m=0}^{\infty} \frac{g^{(m)}(a)}{m!} (x-a)^m . \tag{1}$$

Suppose we want to estimate the green value at position x (G_x) at which only the blue value (B_x) is known. Let's consider the 1-D case with an edge boundary on the right-hand side of B_x, as shown in Fig. 3. Since the nearest known green value is one pixel away, we apply Taylor series to extrapolate the green value pixel at position x along the edge from pixels on the left-hand side of it

$$g(x) = g(x-1) + g'(x-1) + \frac{1}{2!}g''(x-1) + \cdots + \frac{1}{n!}g^{(n)}(x-1) + \cdots \tag{2}$$

where $g(x)$ is the value of a green pixel at location x. For first-order approximations, we assume that

$$g^{(n)}(x-1) = 0 \ \text{ for } \ n \geq 2 \tag{3}$$

and using the central difference approximation for first-order differentiation

$$g'(x-1) = \frac{g(x) - g(x-2)}{2} = \frac{\bar{G}_x - \bar{G}_{x-2}}{2} \tag{4}$$

where $\bar{G}_x$ and $\bar{G}_{x-2}$ are the missing green pixel values at position x and $x-2$, respectively. Different results may be obtained by using forward and/or backward difference approximations.

Based on the assumption that the green and red/blue pixel values are well correlated with constant offsets [1], [4], [23], along an edge, we can approximate the color differences as follows:

$$\bar{G}_x - B_x \approx G_{x-1} - \bar{B}_{x-1} \approx \bar{G}_{x-2} - B_{x-2}. \tag{5}$$

From (5)

$$\bar{G}_x - \bar{G}_{x-2} = B_x - B_{x-2}. \tag{6}$$

Substituting (6) into (4)

$$g'(x-1) = \frac{B_x - B_{x-2}}{2} \tag{7}$$

Hence, by (2), (3) and (7)

$$\hat{G}_x = G_{x-1} + \frac{1}{2}(B_x - B_{x-2}) \tag{8}$$

where $\hat{G}_x$ is the estimated green value at position x. By using Taylor series, we have derived the same (8) as given by Lu and Tan [23].

We can improve the accuracy of the green plane approximation by including the next higher order term, i.e., $g''(x-1)$,

into the equation. For second-order approximation, we assume the third and higher order derivatives are negligible

$$g^{(n)}(x-1) = 0 \; for \; n \geq 3 \tag{9}$$

then (2) can be approximated as

$$\hat{G}_x = G_{x-1} + g'(x-1) + \frac{1}{2}g''(x-1). \tag{10}$$

By applying the central difference approximation for $g''(x-1)$, we obtain

$$g''(x-1) = \frac{g'(x) - g'(x-2)}{2}. \tag{11}$$

Hence, (10) can be modified to:

$$\hat{G}_x = G_{x-1} + g'(x-1) + \frac{1}{4}(g'(x) - g'(x-2)). \tag{12}$$

Using the central difference approximation for $g^{(3)}(x-1)$, we obtain

$$g^{(3)}(x-1) = \frac{g''(x-\frac{1}{2}) - g''(x-\frac{3}{2})}{(x-\frac{1}{2}) - (x-\frac{3}{2})}. \tag{13}$$

By (9), this implies that

$$g''\left(x-\frac{1}{2}\right) - g''\left(x-\frac{3}{2}\right) = 0. \tag{14}$$

Using central difference approximation for $g''(x-1/2)$ and $g''(x-3/2)$, we have

$$(g'(x) - g'(x-1)) - (g'(x-1) - g'(x-2)) = 0 \tag{15}$$

$$\therefore g'(x) = 2g'(x-1) - g'(x-2). \tag{16}$$

Substituting (16) into (12)

$$\hat{G}_x = G_{x-1} + \frac{3}{2}g'(x-1) - \frac{1}{2}g'(x-2). \tag{17}$$

Applying the central difference approximation

$$g'(x-1) = \frac{g(x) - g(x-2)}{2} = \frac{\bar{G}_x - \bar{G}_{x-2}}{2} \tag{18}$$

and

$$g'(x-2) = \frac{g(x-1) - g(x-3)}{2} = \frac{G_{x-1} - G_{x-3}}{2}. \tag{19}$$

Substituting (18) and (19) into (17) gives

$$\hat{G}_x = G_{x-1} + \frac{3}{4}(\bar{G}_x - \bar{G}_{x-2}) - \frac{1}{4}(G_{x-1} - G_{x-3}). \tag{20}$$

Hence, using (6), (20) becomes

$$\hat{G}_x = G_{x-1} + \frac{3}{4}(B_x - B_{x-2}) - \frac{1}{4}(G_{x-1} - G_{x-3}). \tag{21}$$

Now $\hat{G}_x$ can be determined more accurately by including more surrounding known color values in the Bayer pattern. For a 2-D image, we can obtain the extrapolated values for the other three directions, right, top, and bottom directions similarly.

Fig. 1 shows an 8×8 window of a Bayer array neighborhood, where the index (i, j) of each color is given by the row and column location. Consider the case where we want to determine the missing green value $\bar{G}_{45}$. It can be selected from a list of $\left\{\hat{G}^L_{45}, \hat{G}^R_{45}, \hat{G}^T_{45}, \hat{G}^B_{45}\right\}$ where L, R, T and B indicate the left, right, top and bottom directions from which the estimates are extrapolated w.r.t (21)

$$\begin{aligned}
\hat{G}^L_{45} &= G_{44} + \frac{3}{4}(B_{45} - B_{43}) - \frac{1}{4}(G_{44} - G_{42}) \\
\hat{G}^R_{45} &= G_{46} + \frac{3}{4}(B_{45} - B_{47}) - \frac{1}{4}(G_{46} - G_{48}) \\
\hat{G}^T_{45} &= G_{35} + \frac{3}{4}(B_{45} - B_{25}) - \frac{1}{4}(G_{35} - G_{15}) \\
\hat{G}^B_{45} &= G_{55} + \frac{3}{4}(B_{45} - B_{65}) - \frac{1}{4}(G_{55} - G_{75}).
\end{aligned} \tag{22}$$

By including the second-order terms, higher accuracy for the estimates of the missing values in the green plane can be achieved.

III. High-Order Interpolation (HOI)

In this section, we describe a new interpolation method which is a variation to the method described in the previous section to determine the four directional estimates of the missing color pixel. The difference between this and the previous section is that we include the adjacent green value on the other side of the missing sample in the interpolation process. This is based on the presumption that the nearest known samples in a 2-D plane to the missing value contains the most accurate information about that missing sample. Including one sample from the other side of the missing sample will produce a more accurate estimate if it is along an edge. However, if it is across an edge, the estimate may not be accurate. However, a robust classifier, such as the one in Section VI, may be able to avoid the selection of it.

In the previous section, an extrapolation method was used to obtained the estimates. It extrapolated in the four directions towards the center, i.e., from left, right, top and bottom, in order to avoid crossing an edge. In this interpolation method, we interpolate across the center to obtain more accurate estimates. By using the central difference approximation in this case, $g'(x)$ and $g'(x-2)$ can be determined as follows:

$$g'(x) = \frac{g(x+1) - g(x-1)}{2} \tag{23}$$

$$g'(x-2) = \frac{g(x-1) - g(x-3)}{2}. \tag{24}$$

For second-order approximation, by substituting (23) and (24) into (11), we have

$$\begin{aligned}
\hat{G}_x = G_{x-1} &+ g'(x-1) \\
&+ \frac{1}{8}(g(x+1) - 2g(x-1) + g(x-3)).
\end{aligned} \tag{25}$$

Combining (25) with (7) gives

$$\begin{aligned}
\hat{G}_x = G_{x-1} &+ \frac{1}{2}(B_x - B_{x-2}) \\
&+ \frac{1}{8}(G_{x+1} - 2G_{x-1} + G_{x-3}).
\end{aligned} \tag{26}$$

For a 2-D image, we can similarly obtain the interpolated values for the other three directions, right, top, and bottom.

The difference between (26) and (21) is that in (26) the weightings for $(B_x - B_{x-2})$ and G_{x-3} are reduced, while counteracted by the introduction of G_{x+1} into the equation. Since both B_{x-2} and G_{x-3} are further away from the sample

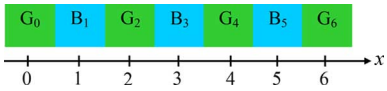

Fig. 4. 1-D Bayer pattern along the x-axis.

to be estimated than G_{x+1}, reducing the weighting of B_{x-2} and G_{x-3} and increasing the weighting of G_{x+1} will produce a more accurate estimate along an edge. This is based on the presumption that the nearer sample contains more accurate information about the sample to be estimated when there is no edge.

Using the same convention as in (22), the new equations for interpolation w.r.t (26) are given as follows:

$$\begin{aligned}
\hat{G}^L_{45} &= G_{44} + \frac{1}{2}(B_{45} - B_{43}) \\
&\quad + \frac{1}{8}(G_{46} - 2G_{44} + G_{42}) \\
\hat{G}^R_{45} &= G_{46} + \frac{1}{2}(B_{45} - B_{47}) \\
&\quad + \frac{1}{8}(G_{44} - 2G_{46} + G_{48}) \\
\hat{G}^T_{45} &= G_{35} + \frac{1}{2}(B_{45} - B_{25}) \\
&\quad + \frac{1}{8}(G_{55} - 2G_{35} + G_{15}) \\
\hat{G}^B_{45} &= G_{55} + \frac{1}{2}(B_{45} - B_{65}) \\
&\quad + \frac{1}{8}(G_{35} - 2G_{55} + G_{75}). \qquad (27)
\end{aligned}$$

Equations (26) and (27) are new interpolation equations derived for more accurate approximation of the green plane.

We have proved in the Appendix that HOI is identical to quadratic spline interpolation. In our next section, we will present cubic spline interpolation as another method in our suite of demosaicking methods.

IV. Cubic Spline Interpolation (CSI)

Cubic spline interpolation is suitable for CFA demosaicking for producing a smooth color image due to the known fact that it can generate a smooth curve to fit a set of data points [22]. Cubic spline interpolation is a piecewise continuous curve, with continuous first and second-order derivatives [6]. A third degree polynomial, $S_i(x)$, is constructed between each point for the given $n+1$ points (x_0, y_0) to (x_n, y_n) and is denoted as the cubic spline

$$S_i(x) = a_i x^3 + b_i x^2 + c_i x + d_i \qquad (28)$$

where i is the index of the spline.

In this paper, we show that further improvement on the results in [22] can be achieved. As shown in the 1-D Bayer pattern in Fig. 4, there are four green data points, and hence $n = 3$. In order to determine the missing green value at position $x = 5$ (i.e., at location B_5), three splines are required as follows:

$$S_0(x) = a_0 x^3 + b_0 x^2 + c_0 x + d_0 \text{ for } x \in [0,2] \qquad (29)$$

$$S_1(x) = a_1 x^3 + b_1 x^2 + c_1 x + d_1 \text{ for } x \in [2,4] \qquad (30)$$

$$S_2(x) = a_2 x^3 + b_2 x^2 + c_2 x + d_2 \text{ for } x \in [4,6]. \qquad (31)$$

From (29)–(31), since there are 12 unknown coefficients: a_0, b_0, c_0, d_0, a_1, b_1, c_1, d_1, a_2, b_2, c_2, and d_2, 12 simultaneous equations are required to obtain a unique solution.

For cubic spline interpolation, three continuity conditions have to be met as follows:

$$\text{1st condition: } S_{i-1}(x_i) = S_i(x_i) \text{ for } i = 1,2 \qquad (32)$$

$$\text{2nd condition: } S'_{i-1}(x_i) = S'_i(x_i) \text{ for } i = 1,2 \qquad (33)$$

$$\text{3rd condition: } S''_{i-1}(x_i) = S''_i(x_i) \text{ for } i = 1,2. \qquad (34)$$

Based on these continuity conditions, there are a total of $4n-2$ equations. In our case for $n = 3$, there are 10 equations as follows.

By the 1st continuity condition (32), we have

$$S_0(0) = G_0 = d_0 \qquad (35)$$

$$S_0(2) = G_2 = 8a_0 + 4b_0 + 2c_0 + d_0 \qquad (36)$$

$$S_1(2) = G_2 = 8a_1 + 4b_1 + 2c_1 + d_1 \qquad (37)$$

$$S_1(4) = G_4 = 64a_1 + 16b_1 + 4c_1 + d_1 \qquad (38)$$

$$S_2(4) = G_4 = 64a_2 + 16b_2 + 4c_2 + d_2 \qquad (39)$$

$$S_2(6) = G_6 = 216a_2 + 36b_2 + 6c_2 + d_2. \qquad (40)$$

The first derivatives of the splines are given by

$$S'_0(x) = 3a_0 x^2 + 2b_0 x + c_0 \qquad (41)$$

$$S'_1(x) = 3a_1 x^2 + 2b_1 x + c_1 \qquad (42)$$

$$S'_2(x) = 3a_2 x^2 + 2b_2 x + c_2. \qquad (43)$$

By the second continuity condition (33), we have

$$S'_0(2) = S'_1(2) \qquad (44)$$

$$\therefore 12a_0 + 4b_0 + c_0 - 12a_1 - 4b_1 - c_1 = 0 \qquad (45)$$

$$S'_1(4) = S'_2(4) \qquad (46)$$

$$\therefore 48a_1 + 8b_1 + c_1 - 48a_2 - 8b_2 - c_2 = 0. \qquad (47)$$

The second derivatives of the splines are given by

$$S''_0(x) = 6a_0 x + 2b_0 \qquad (48)$$

$$S''_1(x) = 6a_1 x + 2b_1 \qquad (49)$$

$$S''_2(x) = 6a_2 x + 2b_2. \qquad (50)$$

By the third continuity condition (34), we have

$$S''_0(2) = S''_1(2) \qquad (51)$$

$$\therefore 12a_0 + 2b_0 - 12a_1 - 2b_1 = 0 \qquad (52)$$

$$S''_1(4) = S''_2(4) \qquad (53)$$

$$\therefore 24a_1 + 2b_1 - 24a_2 - 2b_2 = 0. \qquad (54)$$

The estimates of the green values at positions $x = 1, 3$, and 5 are given by

$$\hat{G}_1 = S_0(1) = a_0 + b_0 + c_0 + d_0 \qquad (55)$$

$$\hat{G}_3 = S_1(3) = 27a_1 + 9b_1 + 3c_1 + d_1 \quad (56)$$

$$\hat{G}_5 = S_2(5) = 125a_2 + 25b_2 + 5c_2 + d_2. \quad (57)$$

As 12 equations are needed to obtain a unique solution, two extra conditions are required. Using the hue assumption [23] based on the correlation between color planes, we propose the extra two equations be derived as follows. By (6), this implies

$$B_3 - B_1 = \bar{G}_3 - \bar{G}_1. \quad (58)$$

Hence, by (55) and (56)

$$B_3 - B_1 = 27a_1 + 9b_1 + 3c_1 + d_1 - a_0 - b_0 - c_0 - d_0. \quad (59)$$

Similarly

$$B_5 - B_3 = \bar{G}_5 - \bar{G}_3. \quad (60)$$

Hence, by (56) and (57)

$$B_5 - B_3 = 125a_2 + 25b_2 + 5c_2 + d_2 - 27a_1 - 9b_1 - 3c_1 - d_1. \quad (61)$$

To solve the 12 simultaneous equations, namely (35)–(40), (45), (47), (52), (54), (59), and (61), the matrix representation (62) is used

$$MC = V \quad (62)$$

where M, C and V are given by (63)–(65). See equations (63)–(66), shown at the bottom of the page, where M^{-1} is the inverse of the matrix M. It can be readily shown that this inverse exists and can be pre-evaluated and stored. Hence, this makes the algorithm computationally efficient. In our case, the

$$\text{Let } M = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 8 & 4 & 2 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 8 & 4 & 2 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 64 & 16 & 4 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 64 & 16 & 4 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 216 & 36 & 6 & 1 \\ 12 & 4 & 1 & 0 & -12 & -4 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 48 & 8 & 1 & 0 & -48 & -8 & -1 & 0 \\ 12 & 2 & 0 & 0 & -12 & -2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 24 & 2 & 0 & 0 & -24 & -2 & 0 & 0 \\ -1 & -1 & -1 & -1 & 27 & 9 & 3 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -27 & -9 & -3 & -1 & 125 & 25 & 5 & 1 \end{bmatrix} \quad (63)$$

$$C = \begin{bmatrix} a_0 \\ b_0 \\ c_0 \\ d_0 \\ a_1 \\ b_1 \\ c_1 \\ d_1 \\ a_2 \\ b_2 \\ c_2 \\ d_2 \end{bmatrix} \quad (64)$$

$$V = \begin{bmatrix} G_0 \\ G_2 \\ G_2 \\ G_4 \\ G_4 \\ G_6 \\ 0 \\ 0 \\ 0 \\ 0 \\ B_3 - B_1 \\ B_5 - B_3 \end{bmatrix} \quad (65)$$

$$\therefore C = M^{-1}V \quad (66)$$

estimate on the left hand side of the missing value is evaluated at $x = 5$ by (57). Hence

$$\hat{G}_5 = \frac{1}{48}\{G_0 + 23G_2 + 23G_4 + G_6 + 8(B_3 - B_1) + 40(B_5 - B_3)\}. \quad (67)$$

In general

$$\hat{G}_x = \frac{1}{48}\{G_{x+1} + 23G_{x-1} + 23G_{x-3} + G_{x-5} + 8(B_{x-2} - B_{x-4}) + 40(B_x - B_{x-2})\}. \quad (68)$$

The other three estimates (R, T and B) can be determined similarly. Likewise, a missing green value at a red pixel position can be evaluated using similar equations.

For comparing our above proposed CSI method with the one given in [22], a general equation for the missing green pixel value at position x can be obtained by solving the equations given in [22] as follows:

$$\hat{G}_x = \frac{1}{48}\{G_{x+3} + 23G_{x+1} + 23G_{x-1} + G_{x-3} + 8(B_x - B_{x-2}) - 8(B_{x+2} - B_x)\}. \quad (69)$$

Clearly, (69) is different from (68) for the missing green value for our proposed CSI method in this paper. The comparison in performance of these two CSI methods will be presented in Section VIII.

V. Interpolation for the Red and Blue Planes

In most cases, higher order approximation is required for the green plane only, and first-order approximation is adequate for the red and blue planes. This is because the human visual system is less sensitive to red and blue [2], [4], and, hence, first-order interpolation is sufficient. It should also be noted that the red and blue pixels are severely undersampled and thus no more improved accuracy can be obtained by using higher order interpolation. Equations (70) and (71) are the equations for determining the red pixel value at a blue and green position, respectively. Equations are similar for the blue plane

$$\begin{aligned}
\hat{R}_{45}^{TL} &= R_{34} + (\hat{G}_{45} - \hat{G}_{34}) \\
\hat{R}_{45}^{TR} &= R_{36} + (\hat{G}_{45} - \hat{G}_{36}) \\
\hat{R}_{45}^{BL} &= R_{54} + (\hat{G}_{45} - \hat{G}_{54}) \\
\hat{R}_{45}^{BR} &= R_{56} + (\hat{G}_{45} - \hat{G}_{56}).
\end{aligned} \quad (70)$$

$$\begin{aligned}
\hat{R}_{44}^{L} &= R_{43} + (G_{44} - \hat{G}_{43}) \\
\hat{R}_{44}^{R} &= R_{45} + (G_{44} - \hat{G}_{45}) \\
\hat{R}_{44}^{T} &= R_{34} + (G_{44} - \hat{G}_{34}) \\
\hat{R}_{44}^{B} &= R_{54} + (G_{44} - \hat{G}_{54}).
\end{aligned} \quad (71)$$

VI. Edge Orientation Classifier and the Weighted Median Filter

We describe in this section the output stage of our proposed demosaicking method. An edge orientation map is used as a classifier to determine the filter weights for a weighted median filter, which is used instead of a linear combiner in this stage. Since a linear combiner, as used in other methods [12], [23], will blur an edge, our method will preserve sharp edges better. Obviously, our edge orientation classifier is not critical in its performance when there is no edge. In this case, any of the four estimates will give a good result and the performance will not depend on the classifier selection.

In our classification process, an edge orientation map for every pixel is produced using the CFA image input. This is used to indicate the possible orientation of an edge for that pixel. For the pixel to be interpolated, the horizontal and vertical gradients are determined by the known neighboring pixels in the CFA input. The gradient is large when crossing an edge. In other words, the direction in which the gradient is smaller is assumed to be the direction along the edge in the neighborhood of this pixel.

At a red/blue pixel, e.g., B_{45}, we define the vertical gradient and horizontal gradient as

$$V = |G_{35} - G_{55}|, \; H = |G_{44} - G_{46}|. \quad (72)$$

At a green pixel, e.g., G_{44}, the vertical and horizontal gradients are defined as

$$V = |R_{34} - R_{54}|, \; H = |B_{43} - B_{45}|. \quad (73)$$

A logical function is used to produce an edge orientation map for the whole color filter array image

$$f(V < H) = \begin{cases} 1, & \text{if } V < H \\ 0, & \text{otherwise.} \end{cases} \quad (74)$$

A "1" in the orientation map indicates that a possible edge along the vertical direction exists at that position while a "0" indicates a possible edge along the horizontal direction.

To estimate the red/blue pixels at blue/red pixel locations, an edge orientation map for diagonal edges is used instead, due to the known red and blue pixel locations in the Bayer pattern. For example at B_{45}, the diagonal gradients are $|R_{34} - R_{56}|$ and $|R_{36} - R_{54}|$ instead of $|G_{35} - G_{55}|$ and $|G_{44} - G_{46}|$ in (72), respectively.

Each edge orientation map is passed through a 2-D standard median filter to remove any outliers. A weighted median filter (WMF) is then used to produce an output based on the edge orientation map. As the edge orientation map only gives a rough estimation of the edge orientation, the two estimates from the other orientation should not be completely excluded because otherwise, errors from the edge orientation map may manifest themselves as color artifacts in the demosaicked output. The edge orientation map output will be used to assign a higher weighting of the estimates for the orientation indicated by the map and in turn increase their probability to appear at the WMF output.

Weighted median filters inherit the robustness and edge preserving capability of the median filters. However, they offer much greater flexibility in design specifications than median filters [30] and the choice of weights allows a certain degree of flexibility in preserving signal structures [26]. The weights of the WMFs control the filtering behavior. For a discrete-time vector $\underline{X} = [X_1, X_2, \ldots, X_N]$, the output Y of the WMF of width N associated with the filter weights

$$\underline{W} = [W_1, W_2, \ldots, W_N] \quad (75)$$

is given by

$$Y = \text{MEDIAN}[W_1 \diamond X_1, W_2 \diamond X_2, \ldots, W_N \diamond X_N] \quad (76)$$

where $\text{MEDIAN}[.]$ denotes the median operation and $\diamond$ denotes the duplication operator, i.e.,

$$K \diamond X = \overbrace{X, \ldots, X}^{K \text{ times}} \quad (77)$$

where K is the duplication number.

The weighted median filtering process sorts the samples within the filter window, duplicates each sample according to the number of the corresponding weight and then selects the median value from the new sequence. Hence, the weights of the WMF influence the probability of a particular sample to be chosen as the output.

By utilising the edge information in the orientation map, the weights for each of the four estimates can be adjusted accordingly. In this case, a WMF with window width of 4 is used. For 1-D signal processing, the sum of weights of a WMF is always an odd number in order to preserve an edge, and in this case only one set of filter weights exists, namely $[2, 1, 1, 1]$, excluding weight permutations [30]. Its distribution function is identical to that of a median filter of width 3 [31]. However, for 2-D image processing in our case, the restriction can be relaxed to allow the sum of the weights for the WMF to be an even number, without blurring an edge. This is because there are at least two samples on the same side of an edge. Moreover, the weights for $\hat{G}^L$ and $\hat{G}^R$ should be equal because there is no extra information from the edge orientation map to distinguish between them, and so for $\hat{G}^T$ and $\hat{G}^B$. In this case, the only possible weight is $[2, 2, 1, 1]$ excluding weight permutations.

Fig. 5 gives the probability density functions of the WMFs of width four with weights $[1, 1, 1, 1]$, $[2, 1, 1, 1]$ and $[2, 2, 1, 1]$ for uniformly distributed samples, as well as the distribution of a 4-point mean for comparison. This gives some insight into the general statistical properties of the weighted median filters. In Fig. 5, sample values at both ends of the x-axis represent outliers. As all three curves of the WMFs drop to zero at both ends, this indicates that all three WMFs can remove outliers very well. For the 4-point mean curve, it is more densely distributed around the mean value. This implies that blurring is more likely to occur. On the other hand, the broader distributions of the WMFs indicate that values other than the mean are more likely to pass through the filter. Hence, the WMFs will produce a sharper image than the output produced by the 4-point mean.

Correspondingly, for a vertical edge indicated in the edge orientation map, the output is given by

$$\hat{G} = \text{MEDIAN}[\hat{G}^L, \hat{G}^R, 2 \diamond \hat{G}^T, 2 \diamond \hat{G}^B]. \quad (78)$$

Similarly, for a horizontal edge indicated in the edge orientation map, the output is given by

$$\hat{G} = \text{MEDIAN}[2 \diamond \hat{G}^L, 2 \diamond \hat{G}^R, \hat{G}^T, \hat{G}^B]. \quad (79)$$

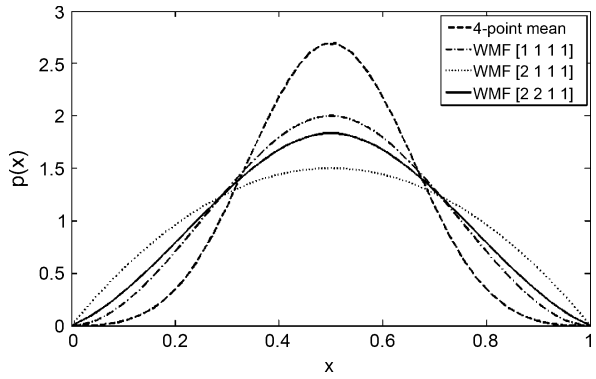

Fig. 5. Density functions of weighted median filters of width 4 with three possible sets of weights and 4-point mean for uniformly distributed samples.

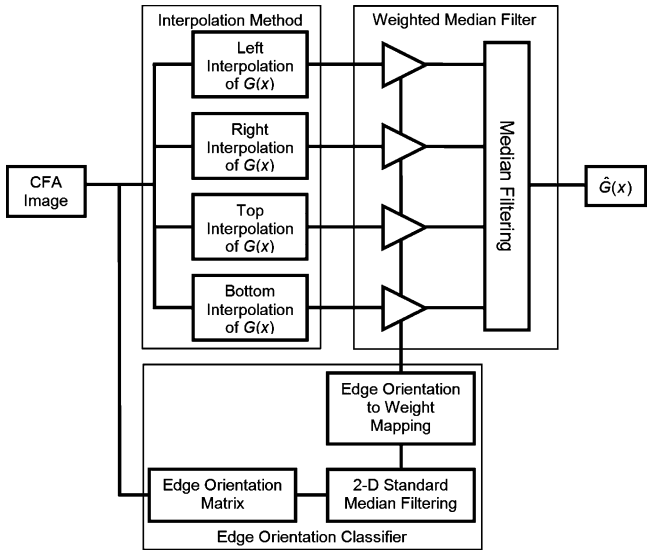

Fig. 6. Block diagram of the proposed algorithm for the green plane.

VII. Proposed Demosaicking Algorithm

Fig. 6 shows the block diagram of our proposed algorithm for the green plane. Four estimates for the missing green pixel, one from each direction, namely left, right, top, and bottom, will first be interpolated using the methods described previously. The four estimates will then be sent to the weighted median filter to determine the green estimate output. The weights used are determined by the edge orientation map according to (78) and (79). The edge orientation map output is passed through a 2-D standard median filter to remove outliers before the edge orientation to weight mapping is carried out. A square window of 7×7 for the 2-D standard median filter will suffice for the images used experimentally in the next section of this paper.

Similarly, the proposed algorithm in Fig. 6 can be applied to the red and blue planes. Since the red and blue planes are only sampled at half the frequency of the green plane, a first-order interpolation is adequate to determine the missing red and blue pixel values.

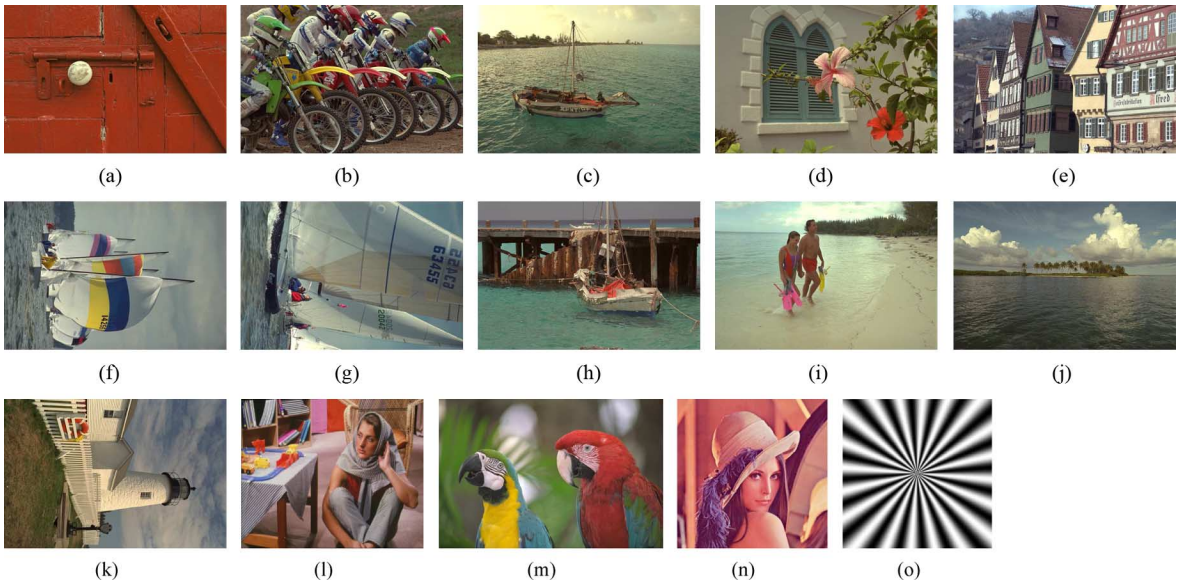

Fig. 7. Images used for performance evaluation.

VIII. Experimental Results

To assess the performance of our proposed methods, namely high order interpolation (WM-HOI), high order extrapolation (WM-HOX), and cubic spline interpolation (WM-CSI) with weighted median filters, we compared them to other prominent demosaicking techniques, namely bilinear interpolation, Freeman's method [7], Kimmel's method [12], Hamilton's method [9], Lu&Tan's method [23], Gunturk's method [8], Plataniotis' method [25], Hirakawa's method [11], Li's method [16], Li&Randhawa's high-order extrapolation (M-HOX) [17], and Li&Randhawa's Cubic Spline Interpolation (CSI) [22]. The difference between the M-HOX method in [17] and our proposed WM-HOX described in this paper is that the former method uses a median filter while the latter uses a weighted median filter as the classifier. Our proposed WM-CSI method in this paper adopts a different approach to that of the CSI method in [22]. The CSI method [22] interpolates only along either the vertical or horizontal direction based purely on the edge orientation map. This differs from our proposed WM-CSI method in that four estimates were interpolated using cubic spline with a weighted median filter as the classifier.

Fifteen images representing different types of images with different characteristics, as shown in Fig. 7, were used for our performance assessment. For example, image (k), the Lighthouse, is a benchmark image having a picket-fence region with vertical edges closely spaced together and this is a challenge to most demosaicking methods. Other images also contain other challenging features such as the red door in image (a) which contains a large region of monotonic color with fine details in the grain of timber, the skin color in image (n), the fine colorful feathers of the parrots in image (m), the water ripples in image (c), and sharp color edges in the sail of image (f). We also included one simulated image, namely the starburst in image (o).

In order to analyse the various demosaicking methods quantitatively, Tables I and II give the Peak Signal-to-Noise Ratio (PSNR) [3] and Normalized Color Difference (NCD) [24] of the various demosaicing methods. PSNR does not always reflect the actual human perception of image quality because it is based on the log ratio of the signal to noise in RGB color space. Hence, in addition we used NCD as a measure of the similarity of the original image and the demosaicked output in the LUV color space because the color difference between two points in the LUV color space has a linear relationship with the perceptual color difference in human vision [24]. In Tables I and II, our methods were ranked the best for the majority of the test images, or otherwise our methods were ranked at least second best. In comparing the results of the methods in [17] and [22] with our proposed corresponding methods in this paper, this demonstrates that the weighted median filter is a superior classifier.

Fig. 8 shows the demosaicked results produced by the various demosaicking methods for the picket-fence region of the Lighthouse image. This region was used for the assessment of the performance on color edge preservation of our demosaicking methods because it contains vertical edges closely spaced together. Various demosaicking methods produced various degrees of color artifacts in this region. However, all our methods, namely M-HOX, M-HOI, CSI, WM-HOX, WM-HOI and WM-CSI, produced negligible amount of visual color artifacts and this demonstrates that our methods are able to reproduce color with high accuracy even at sharp edges.

To further assess the accuracy of our interpolation methods, namely HOX, HOI and CSI, without the influence of the classifier and the weighted median filter in the output stage, we use the original image to select one of the four estimates for each pixel based on the absolute error difference. In Table III, the best choice is one of the four estimates with the minimum absolute error difference between the original pixel. Table III gives the performance indices of the proposed interpolation methods using PSNR and NCD. It shows the best attainable result which is the upper limit of the performance of our selector for the corresponding interpolation method. In other words, it indicates the degree of accuracy that our interpolation methods can achieve.

TABLE I
IMAGE QUALITY PERFORMANCE MEASURES—PSNR dB

Image	(a)	(b)	(c)	(d)	(e)	(f)	(g)	(h)	(i)	(j)	(k)	(l)	(m)	(n)	(o)
Bilinear	33.16	26.71	27.85	33.54	23.58	32.49	32.51	29.18	33.54	31.35	28.10	26.94	36.22	33.19	27.19
Freeman [7]	38.10	34.19	33.51	40.20	28.61	38.18	38.94	35.17	39.00	36.71	33.43	31.56	42.27	34.51	35.69
Kimmel [12]	33.37	29.31	30.80	36.81	27.02	35.93	35.07	31.90	36.48	34.03	31.30	28.70	38.42	30.03	29.94
Hamilton [9]	37.74	32.79	33.30	39.17	29.81	38.38	38.22	34.49	39.05	36.83	34.72	30.50	41.51	34.71	32.29
LuTan [23]	40.50	37.45	37.22	**42.83**	34.47	41.99	42.06	38.70	42.22	40.47	39.43	33.49	44.07	**34.99**	37.17
Gunturk [8]	39.62	37.56	38.83	41.80	35.55	41.92	42.03	39.38	42.71	42.12	40.08	33.55	43.13	33.75	42.66
Plataniotis [25]	34.32	27.44	29.05	33.28	27.30	34.14	33.87	30.83	34.17	32.34	31.41	27.41	33.47	29.24	22.86
Hirakawa [11]	39.22	35.48	37.60	40.62	33.78	40.88	40.60	37.57	41.64	41.50	38.41	**34.27**	42.68	33.81	40.69
XLi [16]	36.53	30.71	31.37	37.46	27.11	36.01	36.15	32.79	36.90	34.74	31.74	29.94	40.10	33.89	31.27
M-HOX [17]	38.04	33.33	36.47	38.26	33.02	39.24	39.34	36.47	39.87	40.11	37.47	32.34	39.32	32.73	43.01
WM-HOX	40.51	36.81	38.77	41.83	35.41	42.27	41.85	39.21	42.73	42.33	39.87	33.64	43.56	33.97	43.22
M-HOI	38.46	34.16	36.94	39.13	33.07	39.73	39.82	36.85	40.44	40.57	37.81	32.53	40.16	33.44	40.66
WM-HOI	**40.87**	**37.65**	38.80	42.73	34.90	**42.46**	**42.21**	39.15	**42.99**	42.36	39.76	33.92	**44.54**	34.65	41.53
CSI [22]	40.08	35.90	38.71	40.72	34.33	41.39	41.19	38.40	42.22	42.24	39.17	32.94	42.85	33.42	42.10
WM-CSI	40.36	36.50	**39.57**	41.00	**35.85**	42.12	41.77	**39.55**	42.64	**43.01**	**40.24**	33.81	42.79	33.41	**47.54**

TABLE II
IMAGE QUALITY PERFORMANCE MEASURES—NCD ($\times 10^{-3}$)

Image	(a)	(b)	(c)	(d)	(e)	(f)	(g)	(h)	(i)	(j)	(k)	(l)	(m)	(n)	(o)
Bilinear	48.55	110.24	60.96	38.57	94.66	25.69	24.49	75.74	23.22	51.04	67.01	76.10	22.48	41.66	53.38
Freeman [7]	31.70	58.62	34.52	22.77	57.03	15.73	14.14	44.29	14.15	30.55	39.12	54.63	15.56	39.15	28.96
Kimmel [12]	48.90	86.91	41.68	27.81	64.59	18.61	17.96	55.96	16.67	35.91	47.24	64.38	20.48	58.64	43.40
Hamilton [9]	33.68	68.57	35.65	24.83	53.15	16.21	15.37	48.05	14.79	30.55	40.71	58.32	16.67	38.40	27.18
LuTan [23]	25.08	42.50	22.41	18.01	32.19	11.57	10.98	29.74	10.53	20.18	24.18	46.58	14.01	**37.63**	23.31
Gunturk [8]	29.01	45.19	20.57	20.64	31.75	12.04	11.39	29.54	10.74	18.11	23.51	49.35	15.08	42.00	19.45
Plataniotis [25]	50.47	135.69	63.63	52.85	80.77	27.32	26.24	74.97	26.28	53.51	59.28	85.05	35.52	64.92	194.43
Hirakawa [11]	26.88	45.36	**18.50**	21.32	29.79	11.79	11.77	**27.01**	10.43	**15.65**	**22.63**	45.90	15.69	41.83	12.88
XLi [16]	35.50	85.92	45.38	29.84	70.66	20.10	18.29	56.82	17.99	37.79	48.99	61.50	18.19	42.19	43.16
M-HOX [17]	33.37	65.47	26.12	26.55	40.28	15.25	14.58	37.88	13.44	22.19	31.21	54.56	21.05	44.64	5.69
WM-HOX	26.13	46.93	20.14	19.64	30.33	11.43	11.39	28.95	10.50	17.37	23.58	48.01	14.77	40.46	4.87
M-HOI	31.23	58.94	24.82	24.70	39.18	14.69	13.83	36.26	12.72	21.11	30.23	51.00	19.57	42.70	4.60
WM-HOI	**24.41**	**41.49**	19.81	**17.86**	30.55	**11.21**	**10.94**	28.49	**10.13**	17.16	23.72	**44.36**	**13.64**	39.02	3.93
CSI [22]	27.67	52.64	20.69	22.10	32.83	12.45	12.27	31.87	11.29	17.49	25.57	49.88	16.15	42.84	4.03
WM-CSI	26.57	47.94	19.24	21.40	**29.46**	11.68	11.65	27.66	10.66	16.54	22.67	47.76	15.99	42.53	**2.63**

From the results in Table III, the HOI interpolation method has the highest accuracy compared to the HOX and CSI interpolation methods for nine out of fifteen test images we used. This supports the results in Tables I and II that the HOI method when coupled with a WMF, i.e., WM-HOI, can produce better PSNR and NCD results when compared with the other methods. For our proposed WM-CSI interpolation method, it performs well too for some types of images such as the house image (e) and the starburst image (o). For the HOI method, the furthest pixel that will be used in the interpolation is only three pixels away from the missing sample, while for the CSI method, the furthest pixel from the missing sample is five pixels away. As a result, in general the HOI method is better suited for images which contain small features with fine details, such as the fine color feathers of the parrot in image (m), so that the interpolation will avoid including pixels from another feature. The CSI method will perform better for images with larger features.

IX. CONCLUSION

We have proposed a suite of new interpolation techniques based on Taylor series and cubic spline for CFA demosaicking with high accuracy. Using the new interpolation equations derived, four estimates from each opposite direction are determined. A weighted median filter of width four with weights

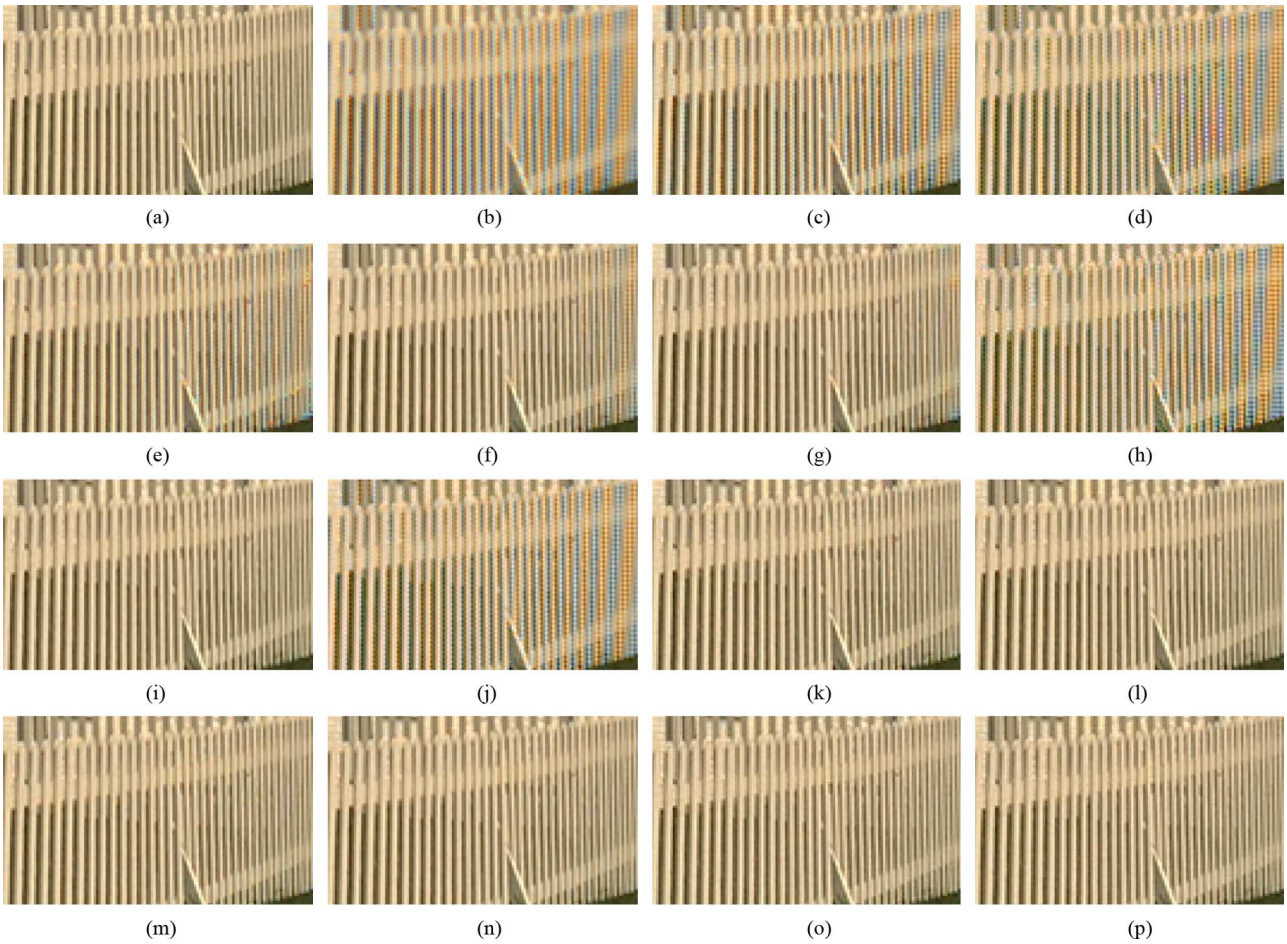

Fig. 8. Picket fence region of (a) the original Lighthouse image and the demosaiced output images using (b) Bilinear interpolation, (c) Freeman, (d) Kimmel, (e) Hamilton, (f) Lu&Tan, (g) Gunturk, (h) Plataniotis, (i) Hirakawa, (j) XLi, (k) M-HOX, (l) WM-HOX, (m) M-HOI, (n) WM-HOI, (o) CSI and (p) WM-CSI.

determined by an edge orientation map is used to produce an output from the four estimates. Our experimental results using various images of different characteristics demonstrated that our techniques can produce a demosaicked image with minimal artifacts and is faithful to the original color even at sharp color edges. There is no outright winner amongst them because they perform differently for various types of images. However, our high-order Interpolation method using the weighted median filter (WM-HOI) will produce better results for most types of images.

APPENDIX
THAT TAYLOR SERIES INTERPOLATION (HOI) IS IDENTICAL TO THE QUADRATIC SPLINE INTERPOLATION

Just like the cubic spline, quadratic spline interpolation is a piecewise continuous curve, with continuous first-order derivatives. Given $n+1$ points (x_0, y_0) to (x_n, y_n), a second degree polynomial, $S_i(x)$, is constructed between each point and is denoted as the quadratic spline

$$S_i(x) = a_i x^2 + b_i x + c_i. \tag{80}$$

Referring to the 1-D horizontal Bayer pattern in Fig. 4, only three green data points (G_0, G_2 and G_4) will be considered for quadratic spline interpolation, with $n = 2$. To determine the missing green value at position $x = 3$ (i.e., at location B_3), two splines are required as follows:

$$S_0(x) = a_0 x^2 + b_0 x + c_0 \text{ for } x \in [0, 2] \tag{81}$$
$$S_1(x) = a_1 x^2 + b_1 x + c_1 \text{ for } x \in [2, 4]. \tag{82}$$

From (81)–(82), since there are six unknown coefficients: a_0, b_0, c_0, a_1, b_1 and c_1, 6 simultaneous equations are required to obtain a unique solution.

For quadratic spline interpolation, two continuity conditions [i.e., (32) and (33)] have to be met. Based on these continuity conditions, there are a total of $4n-2$ equations. In our case for $n = 2$, there are 6 equations as follows. By the 1st continuity condition (32), we have

$$S_0(0) = G_0 = c_0 \tag{83}$$
$$S_0(2) = G_2 = 4a_0 + 2b_0 + c_0 \tag{84}$$
$$S_1(2) = G_2 = 4a_1 + 2b_1 + c_1 \tag{85}$$
$$S_1(4) = G_4 = 16a_1 + 4b_1 + c_1. \tag{86}$$

The first derivatives of the splines are given by

$$S_0'(x) = 2a_0 x^2 + b_0 x. \tag{87}$$
$$S_1'(x) = 2a_1 x^2 + b_1 x. \tag{88}$$

TABLE III
IMAGE QUALITY PERFORMANCE MEASURES FOR THE PROPOSED INTERPOLATION METHODS-PSNR DB AND NCD ($\times 10^{-3}$).

	Image	(a)	(b)	(c)	(d)	(e)	(f)	(g)	(h)	(i)	(j)	(k)	(l)	(m)	(n)	(o)
PSNR	HOX	44.28	40.05	44.17	44.44	41.22	45.67	46.32	43.89	46.30	47.85	45.31	40.15	44.97	38.63	46.55
	HOI	**44.69**	**40.76**	44.74	**45.27**	41.49	**46.07**	**46.80**	44.38	**47.09**	48.49	45.84	**41.55**	**45.84**	**39.09**	52.27
	CSI	44.32	40.30	**45.42**	44.02	**42.05**	45.78	46.64	**44.72**	46.47	**49.01**	**46.10**	39.97	44.37	38.14	**54.70**
NCD	HOX	16.03	31.13	11.64	12.47	18.09	7.16	6.63	16.70	6.11	9.87	13.76	24.37	9.95	23.46	1.82
	HOI	**14.84**	**28.48**	11.00	**11.59**	17.51	**6.81**	**6.23**	16.01	**5.66**	9.28	13.12	**21.34**	**9.07**	**22.28**	1.36
	CSI	15.67	29.17	**10.31**	12.78	**16.75**	6.91	6.30	**15.26**	5.84	**8.79**	**12.57**	24.16	10.43	24.44	**0.94**

By the second continuity condition (33), we have

$$S_0'(2) = S_1'(2) \tag{89}$$

$$\therefore 4a_0 + b_0 - 4a_1 - b_1 = 0. \tag{90}$$

The estimates of the green values at positions $x = 1$ and 3 are given by

$$\hat{G}_1 = S_0(1) = a_0 + b_0 + c_0 \tag{91}$$

$$\hat{G}_3 = S_1(3) = 9a_1 + 3b_1 + c_1. \tag{92}$$

Hence, by (58), (91), and (92)

$$B_3 - B_1 = 9a_1 + 3b_1 + c_1 - a_0 - b_0 - c_0. \tag{93}$$

To solve the 6 simultaneous equations, namely (83)–(86), (90) and (93), the matrix representation (94) is used.

$$MQ = V \tag{94}$$

where M, Q and V are given by (95)–(97)

$$\text{Let } M = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 \\ 4 & 2 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 4 & 2 & 1 \\ 0 & 0 & 0 & 16 & 4 & 1 \\ 4 & 1 & 0 & -4 & -1 & 0 \\ -1 & -1 & -1 & 9 & 3 & 1 \end{bmatrix} \tag{95}$$

$$Q = \begin{bmatrix} a_0 \\ b_0 \\ c_0 \\ a_1 \\ b_1 \\ c_1 \end{bmatrix} \tag{96}$$

$$V = \begin{bmatrix} G_0 \\ G_2 \\ G_2 \\ G_4 \\ 0 \\ B_3 - B_1 \end{bmatrix} \tag{97}$$

$$\therefore Q = M^{-1}V. \tag{98}$$

Hence, the interpolated green value at the blue pixel position at $x = 3$ can be evaluated by (92).

From (95), the inverse of M can be evaluated as follows:

$$M^{-1} = \frac{1}{8}\begin{bmatrix} 3 & 1 & -3 & -1 & 2 & 4 \\ -10 & 2 & 6 & 2 & -4 & -8 \\ 8 & 0 & 0 & 0 & 0 & 0 \\ -1 & -3 & 1 & 3 & 2 & -4 \\ 6 & 18 & -10 & -14 & -12 & 24 \\ -8 & -24 & 24 & 16 & 16 & -32 \end{bmatrix}. \tag{99}$$

By (96)–(99), the coefficients a_1, b_1, and c_1 can be solved as follows:

$$a_1 = \frac{1}{8}[-G_0 - 2G_2 + 3G_4 - 4(B_3 - B_1)] \tag{100}$$

$$b_1 = \frac{1}{8}[6G_8 + G_2 - 14G_4 + 24(B_3 - B_1)] \tag{101}$$

$$c_1 = -G_0 + 4G_4 - 4(B_3 - B_1). \tag{102}$$

By (92)

$$\hat{G}_3 = 9a_1 + 3b_1 + c_1. \tag{103}$$

Substituting (100), (101), and (102) into (103) yields

$$\hat{G}_3 = \frac{1}{8}G_0 + \frac{3}{4}G_2 + \frac{1}{8}G_4 + \frac{1}{2}(B_3 - B_1). \tag{104}$$

Rearranging (104) gives

$$\hat{G}_3 = G_2 + \frac{1}{2}(B_3 - B_1) + \frac{1}{8}(G_4 - 2G_2 + G_0). \tag{105}$$

Without loss of generality for comparison purposes, if we set $x = 3$ in this case, (105) is identical to (26). Thus, we have proved that the quadratic spline interpolation is identical to our proposed HOI method in Section III.

ACKNOWLEDGMENT

The authors would like to thank the anonymous reviewers for their constructive comments and suggestions. The authors would also like to thank B. K. Gunturk and K. Hirakawa for providing their codes.

REFERENCES

[1] J. E. Adams, Jr., "Interactions between color plane interpolation and other image processing functions in electronic photography," *Proc. SPIE*, vol. 2416, pp. 144–151, 1995.
[2] B. E. Bayer, "Color Imaging Array," U.S. patent 3 971 065, 1976.
[3] K. -H. Chung and Y.-H. Chan, "Color demosaicing using variance of color differences," *IEEE Trans. Image Process.*, vol. 15, pp. 2944–2955, 2006.

[4] D. R. Cok, "Signal Processing Method and Apparatus for Producing Interpolated Chrominance Values in a Sampled Color Image Signal," U.S. patent 4 642 678, 1987.
[5] E. R. Davies, "On the noise suppression and image enhancement characteristics of the median, truncated median and mode filters," *Pattern Recognit. Lett.*, vol. 7, pp. 87–97, 1988.
[6] C. De Boor, *A Practical Guide to Splines*. New York: Springer-Verlag, 1978.
[7] W. T. Freeman, "Median Filter for Reconstructing Missing Color Samples," U.S. patent 4,724,395, 1988.
[8] B. K. Gunturk, Y. Altunbasak, and R. M. Mersereau, "Color plane interpolation using alternation projections," *IEEE Trans. Image Process.*, vol. 11, pp. 997–1013, 2002.
[9] J. F. Hamilton, Jr. and J. E. Adams, Jr., "Adaptive Color Plan Interpolation in Single Sensor Color Electronic Camera," U.S. patent 5 629 734, 1997.
[10] R. H. Hibbard, "Apparatus and Method for Adaptively Interpolating a Full Color Image Utilizing Luminance Gradients," U.S. patent 5 382 976, 1995.
[11] K. Hirakawa and T. W. Parks, "Adaptive homogeneity-directed demosaicing algorithm," *IEEE Trans. Image Process.*, vol. 14, pp. 360–369, 2005.
[12] R. Kimmel, "Demosaicing: Image reconstruction from color CCD samples," *IEEE Trans. Image Process.*, vol. 8, pp. 1221–1228, 1999.
[13] E. Kreyszig, *Advanced Engineering Mathematics*. New York: Wiley, 1999.
[14] C. A. Laroche and M. A. Prescott, "Apparatus and Method for Adaptively Interpolating a Full Color Image Utilizing Chrominance Gradients," U.S. patent 5 373 322, 1994.
[15] X. Li and M. T. Orchard, "New edge-directed interpolation," *IEEE Trans. Image Process.*, vol. 10, pp. 1521–1527, 2001.
[16] X. Li, "Demosaicing by successive approximation," *IEEE Trans. Image Process.*, vol. 14, pp. 370–379, 2005.
[17] J. S. J. Li and S. Randhawa, "High order extrapolation using Taylor series for color filter array demosaicing," in *Springer Lecture Notes in Computer Science*. New York: Springer, 2005, pp. 703–711, Series LNCS 3656.
[18] J. S. J. Li and S. Randhawa, "Improved accuracy for colour filter array demosaicing using high order extrapolation," in *Proc. 8th Int. Symp. Signal Processing and its Applications*, 2005, pp. 331–334.
[19] J. S. J. Li and S. Randhawa, "CFA demosaicking by adaptive order of approximation," in *Proc. 1st Int. Conf. Computer Vision Theory and Applications*, 2006, pp. 5–10.
[20] J. S. J. Li and S. Randhawa, "A structural approach to improved colour filter array demosaicking for the Bayer pattern," in *Proc. 8th IASTED Int. Conf. Signal and Image Processing*, 2006, pp. 157–161.
[21] J. S. J. Li and S. Randhawa, "Adaptive CFA demosaicking with mixed order of approximation," in *Proc. IEEE Proc. Information, Decision and Control*, 2007, pp. 326–331.
[22] J. S. J. Li and S. Randhawa, "CFA demosaicking using cubic spline interpolation," in *Proc. IEEE Int. Conf. Acoustics, Speech, and Signal Processing*, 2007, pp. I 865–868.
[23] W. Lu and Y.-P. Tan, "Color filter array demosaicking: New method and performance measures," *IEEE Trans. Image Process.*, vol. 12, pp. 1194–1210, 2003.
[24] K. N. Plataniotis and A. N. Venetsanopoulos, *Color Image Processing and Applications*. New York: Springer Verlag, 2000.
[25] K. N. Plataniotis and R. Lukac, "An efficient demosaicing approach with a global control of correction steps," in *Proc. Int. Conf. Acoustics, Speech, and Signal Processing*, 2004, pp. III 469–472.
[26] M. K. Prasad and Y. H. Lee, "Analysis of weighted median filters based on inequalities relating the weights," *Circuits Syst. Signal Process.*, vol. 11, pp. 115–136, 1992.
[27] W. K. Pratt, *Digital Image Processing*, 3rd ed. New York: Wiley, 2001.
[28] S. Randhawa and J. S. J. Li, "CFA demosaicing with improved colour edge preservation," presented at the IEEE Int. Region 10 Conf., 2005.
[29] M. A. Wober and R. Soini, "Method and Apparatus for Recovering Image Data Through the Use of a Color Test Pattern," U.S. patent 5 475 769, 1995.
[30] L. Yin, R. Yang, M. Gabbouj, and Y. Neuvo, "Weighted median filters: A tutorial," *IEEE Trans. Circuits Syst.*, vol. 43, pp. 157–192, 1996.
[31] O. Yli-Harja, J. Astola, and Y. Neuvo, "Analysis of the properties of median and weighted median filters using threshold logic and stack filter representation," *IEEE Trans. Signal Process.*, vol. 39, pp. 395–410, 1991.

Jim S. Jimmy Li (M'84) received the B.E. (Hons I) and Ph.D. degrees from the University of New South Wales, Australia, in 1985 and 1989, respectively.

He was with the Chinese University of Hong Kong, University of Canterbury, and DSTO, and is now with Flinders University, School of Computer Science, Engineering and Mathematics. He was the Publicity Chair for IDC 2002.

Sharmil Randhawa (S'97–M'09) received the B.E. (Hons) degree from the University of Adelaide, Australia, and M.Eng. (Biomedical) degree from Flinders University, Australia, in 1993 and 1997, respectively, where she is currently pursuing the Ph.D. degree.

She is with the School of Computer Science, Engineering and Mathematics, Flinders University. She is a member of the IEEE SA section Women in Engineering committee, and is on the organizing committee for AAEE 2009.